RAG-Based Medical Diagnostic System

A sophisticated diagnostic assistant combining workflow automation, vector-based semantic retrieval, and generative AI to deliver accurate medical insights in real time.

System Architecture Overview

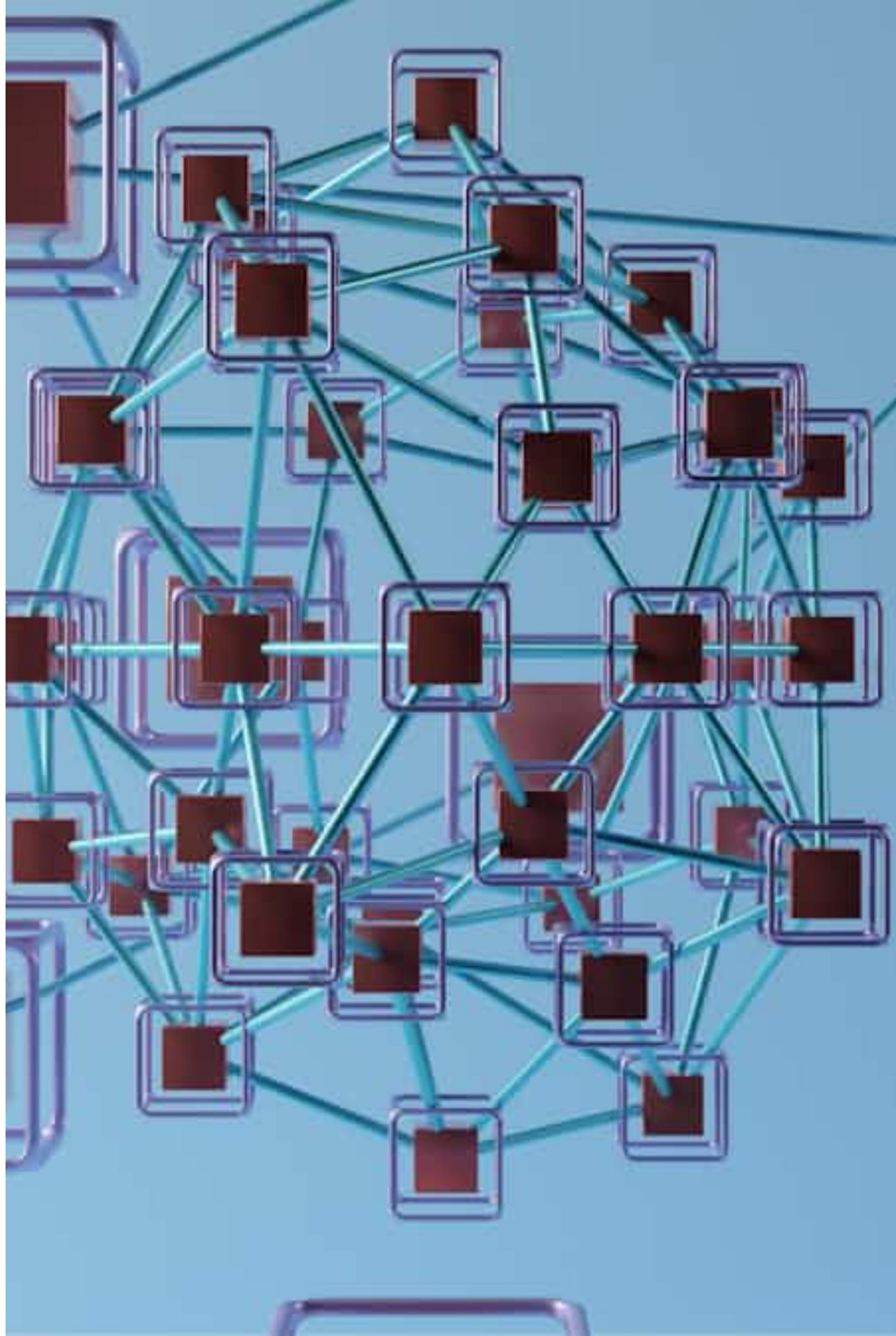
Ingestion Pipeline

Processes medical data and converts it into vector embeddings stored in Supabase for semantic search.

Retrieval Pipeline

Executes when users enter symptoms, retrieving similar entries and generating diagnostic recommendations via LLM.

Built on n8n for workflow automation, Supabase for vector search, and Node.js/Vite for the user interface—ensuring seamless communication between frontend, automation engine, and AI components.



Data Foundation

Structured Dataset

The system relies on a standardized CSV file containing cleaned and preprocessed medical data.

This consistent schema enables efficient semantic search during retrieval.

Disease

Medical condition name

Symptoms

Common symptoms associated with the disease

Diagnostic Measures

Recommended medical tests for confirmation

Ingestion Pipeline: Data to Embeddings

CSV Input

Preprocessed medical data loaded into workflow

Embedding Generation

Text converted to dense vector representations

Vector Storage

Embeddings stored in Supabase pgvector

n8n orchestrates the entire workflow in Docker, ensuring consistency during embedding and enabling fast semantic search in the retrieval stage.

Ingestion Components

n8n Automation

Dockerized workflow engine orchestrating ingestion processes and managing component connections.

Embedding Model

Converts disease and symptom text into semantic vectors capturing medical meaning.

Supabase pgvector

Stores embeddings and original text for fast, accurate similarity searches.

User Journey: From Symptoms to Diagnosis

1

User Input

Patient enters symptoms via web chatbot interface

2

Webhook Trigger

Node.js backend sends input to n8n workflow

3

Vector Search

Query embedded and matched against database

4

LLM Generation

AI produces diagnostic recommendation

5

Response Display

Result shown to user in natural language

Retrieval Pipeline Deep Dive

01

Query Embedding

User symptoms converted to vector using same model from ingestion stage

02

Semantic Search

k-NN search in Supabase retrieves top matching symptom patterns

03

Context Compilation

Retrieved entries (Disease, Symptoms, Diagnostic Measures) assembled into context block

04

LLM Processing

Context and query passed to LLM via structured prompt for diagnosis generation

05

Response Delivery

Generated text returned through webhook to frontend for display

Technical Stack

Data Layer

Structured CSV (preprocessed.csv) with diseases, symptoms, and diagnostics

Automation

n8n running in Docker for workflow orchestration

Vector Database

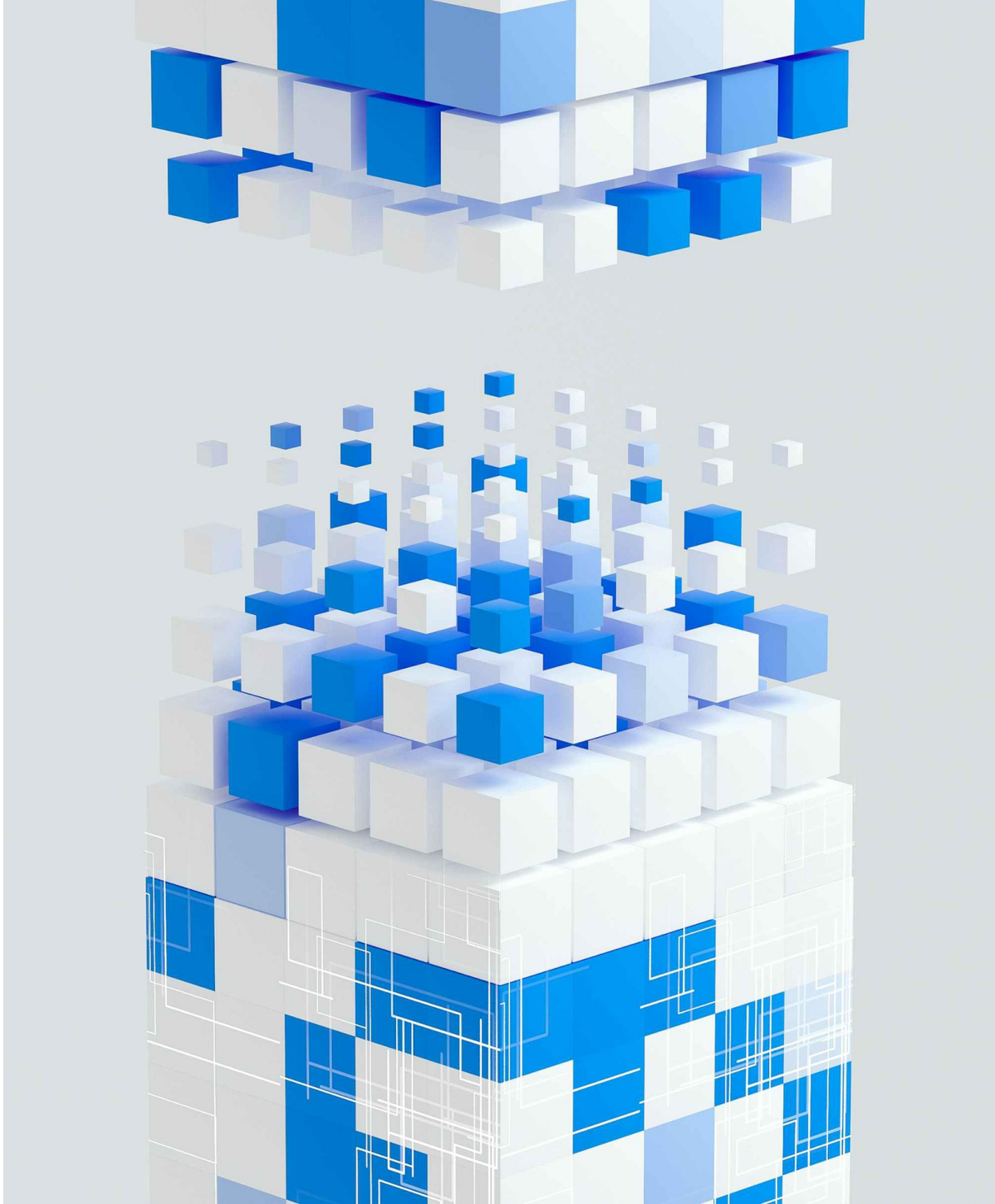
Supabase with pgvector extension for semantic similarity searches

Frontend

Node.js and Vite for user interaction and API communication

AI Components

Embedding model and LLM for semantic understanding and generation



System Benefits

Fast Response

Real-time diagnostic insights delivered through optimized vector search and AI generation.

Contextually Intelligent

Semantic understanding ensures recommendations are grounded in relevant medical knowledge.

Scalable Architecture

Docker containerization and cloud database enable easy expansion and maintenance.

The Complete Picture

This RAG-based system combines workflow automation, vector-based semantic retrieval, and generative AI to deliver medical diagnostic insights.

- Ingestion pipeline ensures data is properly structured and embedded
- Retrieval pipeline dynamically interprets user symptoms
- System produces contextually grounded, human-readable diagnostic recommendations